

55-64  
- 1-97565  
N 94-21469

TIME-ACCURATE NAVIER-STOKES CALCULATIONS  
WITH MULTIGRID ACCELERATION

N. Duane Melson  
NASA Langley Research Center  
Hampton, VA 23681-0001

Mark D. Sanetrik  
Analytical Services and Materials, Incorporated  
Hampton, VA 23681-0001

Harold L. Atkins  
NASA Langley Research Center  
Hampton, VA 23681-0001

### SUMMARY

A numerical scheme to solve the unsteady Navier-Stokes equations is described. The scheme is implemented by modifying the multigrid-multiblock version of the steady Navier-Stokes equations solver, TLNS3D. The scheme is fully implicit in time and uses TLNS3D to iteratively invert the equations at each physical time step. The design objective of the scheme is unconditional stability (at least for first- and second-order discretizations of the physical time derivatives). With unconditional stability, the choice of the time step is based on the physical phenomena to be resolved rather than limited by numerical stability which is especially important for high Reynolds number viscous flows, where the spatial variation of grid cell size can be as much as six orders of magnitude.

An analysis of the iterative procedure and the implementation of this procedure in TLNS3D are discussed. Numerical results are presented to show both the capabilities of the scheme and its speedup relative to the use of global minimum time stepping. Reductions in computational times of an order of magnitude are demonstrated.

### INTRODUCTION

Although significant progress has been made in the last twenty years to numerically model many physical situations, most numerical schemes are limited to the prediction of steady flows. This limitation is particularly true in the field of computational fluid dynamics (CFD), where solutions to the Navier-Stokes equations for steady flows are now calculated on a regular basis. (See, for example, references [1-3]). An important factor that has lead to the increased use of Navier-Stokes solvers is the recent success in

reducing the computer resources necessary to obtain converged solutions. Perhaps the most promising work has been in the use of multigrid acceleration techniques. Convergence to steady state has been shown in  $O[\log(n)]$  work, where  $n$  represents the number of unknowns to be solved. This reduction in computer requirements has made steady-state solutions affordable to the practicing engineer.

However, many physical phenomena (e.g., separated flows, wake flows, buffet) are intrinsically unsteady. The solution of unsteady problems in CFD has been limited to simplified subsets of the Navier-Stokes equations (panel methods, potential-flow solvers, and some limited use of Euler equation solvers). Unsteady Navier-Stokes calculations have been too expensive for routine use.

The present approach is to apply an iterative procedure for the solution of an implicit equation; thus, the approach is called an *iterative-implicit* method. The concept is not new; in fact, many of the methods developed in the field of linear algebra for inverting large matrices are iterative. Within the field of CFD, similar work is discussed by Jameson [4] for unsteady flows and by Taylor, Ng, and Walters [5] for steady-state flows. The present approach is similar to that of Jameson in that a Runge-Kutta based multigrid method is used to solve the implicit unsteady flow equations. The Navier-Stokes equations have been treated in the present work, and Jameson's implementation has been modified so that the robustness of the scheme is dramatically increased.

A detailed description of the implementation will be followed by an analysis of the method and the numerical results from one- and two-dimensional test problems.

## TIME-DEPENDENT METHOD

In the present work, a modified version of the thin-layer Navier-Stokes (TLNS) equations is used to model the flow. The acronym "TLNS" used here describes an equation set obtained from the complete Reynolds-averaged Navier-Stokes equations by retaining only the viscous diffusion terms normal to the solid surfaces. The effects of turbulence are modeled through an eddy-viscosity hypothesis. The Baldwin-Lomax turbulence model [6] is used for turbulence closure. For a body-fitted coordinate system  $(\xi, \eta, \zeta)$  fixed in time, these equations can be written in the conservation-law form as

$$-\frac{\partial}{\partial T}(J^{-1}U) = \frac{\partial F}{\partial \xi} + \frac{\partial G}{\partial \eta} + \frac{\partial H}{\partial \zeta} - \frac{\partial F_v}{\partial \xi} - \frac{\partial G_v}{\partial \eta} - \frac{\partial H_v}{\partial \zeta} \quad (1)$$

where  $U$  represents the conserved variable vector and  $F$ ,  $G$ , and  $H$  represent the convective flux vectors. In the above equation set  $F_v$ ,  $G_v$  and  $H_v$  represent the viscous flux vectors in the three coordinate directions  $(\xi, \eta, \zeta)$ , and  $J$  is the Jacobian of the transformation. These equations represent a more general form of the classical thin-layer equations introduced in reference [6] because the diffusion terms in all three coordinate directions are included in this form. The Euler equations can easily be recovered from equation (1) by simply dropping the last three terms on the right-hand side.

The temporal derivatives are cast as a fully implicit operator in physical time. For first- or second-order discretizations in time, this produces an unconditionally stable scheme, which allows the time-step size to be chosen based on the temporal resolution needed in the solution rather than limited by the numerical

stability requirements. The fully implicit terms are iteratively solved with multigrid acceleration rather than direct inversion, which would be too costly for the nonlinear three-dimensional Navier-Stokes equations.

## IMPLEMENTATION OF TIME-DEPENDENT METHOD

### Original TLNS3D Method

In the original TLNS3D program, a semidiscrete cell-centered finite-volume algorithm, based on a Runge-Kutta time-stepping scheme [1][7][8], is used to obtain the steady-state solutions to the TLNS equations. A linear fourth-difference-based and nonlinear second-difference-based artificial dissipation is added to suppress both the odd-even decoupling and the oscillations in the vicinity of shock waves and stagnation points, respectively. Both the scalar and matrix forms of the artificial dissipation models [9] are incorporated.

In the steady-state implementation, the physical time  $T$  is replaced by a pseudo time  $\tau$ , which gives

$$-\frac{\partial}{\partial \tau}(J^{-1}U) = \frac{\partial F}{\partial \xi} + \frac{\partial G}{\partial \eta} + \frac{\partial H}{\partial \zeta} - \frac{\partial F_v}{\partial \xi} - \frac{\partial G_v}{\partial \eta} - \frac{\partial H_v}{\partial \zeta}. \quad (2)$$

At steady state, the left-hand side of equation (2) disappears, and the right-hand side (the residual) goes to zero, so that any stable scheme may be used to advance the solution in pseudo time.

In the original TLNS3D program, the solution is advanced with a five-stage Runge-Kutta time-stepping scheme. Three evaluations of the artificial dissipation terms (computed at the odd stages) are used to obtain a larger parabolic stability bound, which allows a higher CFL number in the presence of physical viscous diffusion terms. Such a scheme is computationally efficient for solving both the steady Navier-Stokes and the steady Euler equations. The stability range of the numerical scheme is further increased with the use of the implicit residual smoothing technique that employs grid aspect-ratio-dependent coefficients [1][10][11].

Equation (2) can be rewritten to group the convective and diffusive terms from the right-hand side as

$$Vol \frac{\partial(U)}{\partial \tau} + C(U) - D_p(U) - D_a(U) = 0 \quad (3)$$

where the equation has been multiplied through by the volume  $Vol$  and  $C(U)$ ,  $D_p(U)$ , and  $D_a(U)$  are the convection, physical diffusion, and artificial diffusion terms, respectively. The implementation of the Runge-Kutta time stepping is shown by rewriting equation (3) as

$$Vol \frac{U^k - U^0}{\alpha_k \Delta \tau} + C^{k-1}(U) - D_p^0(U) - \widetilde{D_a^{k-1}}(U) = 0 \quad (4)$$

where the superscript  $k$  indicates that the given term should be evaluated at the  $k$ th Runge-Kutta stage. The  $\widetilde{k-1}$  superscript indicates that the terms are evaluated with a linear combination of the values from previous Runge-Kutta stages.

The solution is advanced in pseudo time with the maximum allowable time step for each cell. Enthalpy damping, which has been used in several previous studies to accelerate the convergence of the numerical scheme, is not employed because the Navier-Stokes equations generally do not admit constant enthalpy as a solution. The efficiency of the steady numerical scheme is also significantly enhanced through the use of a multigrid acceleration technique as described in reference [1]. The original TLNS3D program was extensively modified to facilitate solution of the flow fields over a wide range of geometric configurations through domain decomposition. A consequence of this work is the generalization of the boundary conditions of the program to easily accommodate any arbitrary grid topology. A detailed description of this capability is given in reference [12].

### Time-dependent TLNS3D-MB

The physical time derivative of equation (1) is approximated by a discrete operator of the form

$$\frac{\partial U}{\partial T} \approx L_t U^{n+1} \equiv \frac{1}{\Delta T} \left( \sum_{m=0}^M a_m U^{n+1-m} \right) = \frac{1}{\Delta T} \left[ a_0 U^{n+1} + E(U^n, U^{n-1}, \dots, U^{n+1-M}) \right] \quad (5)$$

to give

$$L_t U^{n+1} = S(U^{n+1}) \quad (6)$$

where  $E$  denotes the portion of the discrete physical time derivative that involves values from the previous time steps and  $S(U^{n+1})$  denotes the discrete approximation to the right-hand side of equation (1). Equation (6) is an implicit time-accurate equation for the time advancement of the unsteady solution of the Navier-Stokes equations. The first task is to put equation (6) in a form that is amenable to time-asymptotic steady-state methods such as TLNS3D. This involves construction of an iteration procedure that can be interpreted as a pseudo time. The TLNS3D code employs a Runge-Kutta and multigrid methodology to advance the pseudo time, which introduces an additional level of iteration. To avoid the use of multiple indices and, hopefully, avoid confusion between the various iteration procedures within the overall algorithm, the following change of variables is introduced. Let  $W \equiv U^{n+1}$  and  $W^l$  denote the  $l$ th approximation to  $W$ . Thus,  $\lim_{l \rightarrow \infty} W^l$  is equal to  $U^{n+1}$  whenever the iterative method used for the solution of equation (6) is convergent. In describing the Runge-Kutta scheme, let  $V^k$  denote the solution obtained in the  $k$ th stage of the Runge-Kutta scheme.

Equation (6) is rewritten in this notation to obtain

$$\frac{a_o}{\Delta T} W + \frac{E}{\Delta T} = S(W) \quad (7)$$

where  $E$  again involves the portion of the physical time derivative at previous time steps and is invariant during the iteration process which advances the solution from  $T^n$  to  $T^{n+1}$ . An iterative equation is constructed from equation (6) simply by adding a pseudo-time derivative term to the left-hand side. The only consideration is that the sign of the new term must be the same as that of the physical time derivative.

$$W_\tau + \frac{a_o}{\Delta T} W + \frac{E}{\Delta T} = S(W) \quad (8)$$

When  $\bar{\lambda} \equiv a_0 \Delta \tau / \Delta T$  is small, equation (8) differs from equation (2) by only a small perturbation. Thus, we can reasonably expect that any method that efficiently solves equation (2) would also solve equation (8).

In the previous work of Jameson [4], all contributions from the physical time term were carried to the right-hand side of the equation and treated as explicit terms within the Runge-Kutta stages. With efficiency as a consideration, Jameson suggested the use of this approach only when the CFL, based on the physical time step, is greater than 200. (Note that a large CFL corresponds to a small  $\bar{\lambda}$ ). A later section shows that this explicit approach is actually unstable for large values of  $\bar{\lambda}$ .

In many flows, especially high Reynolds number viscous flows,  $\bar{\lambda}$  may easily become large. In the present work, the Runge-Kutta scheme is made stable for all values of  $\bar{\lambda}$  by treating the contribution of the physical time derivative implicitly within the Runge-Kutta scheme. Because the time derivative appears only as a diagonal term, the modification is easily implemented as follows.

$$\begin{aligned} V^0 &= W^l \\ (1 + \alpha_k \bar{\lambda}) V^k &= V^0 + \alpha_k \Delta \tau \hat{R}^{\overline{k-1}}(V) \quad k = 1, 2, 3, \dots, K \\ W^{l+1} &= V^K \end{aligned} \quad (9)$$

where  $\hat{R}(V) = S(V) - E/\Delta T$  denotes the modified residual, and the superscript  $\overline{k-1}$  denotes that the residual may be a combination of  $\hat{R}(V)$  at all the previous Runge-Kutta stages.

This formulation is not yet appropriate for steady-state flow solvers such as TLNS3D. The reason is that these codes use several acceleration techniques, such as implicit-residual smoothing and multigrid, both of which are designed to operate on a residual term that goes to 0 as the solution converges. Because  $\hat{R}$  contains only the portion of the physical time derivative at previous physical time steps, it converges to  $\bar{\lambda}W$  as  $W^l$  goes to  $W$ . To accommodate the above acceleration techniques,  $\hat{R}$  is rewritten as

$$\Delta \tau \hat{R}(V) = \bar{\lambda}V - \bar{\lambda}V + \Delta \tau \hat{R}(V) = \bar{\lambda}V + \Delta \tau R(V) \quad (10)$$

where

$$R(V) = S(V) - (a_0 V + E)/\Delta T \quad (11)$$

The residual  $R$  contains all the physical terms and goes to 0 as  $W^l$  goes to  $W$ . The Runge-Kutta method, with implicit-residual smoothing and multigrid, becomes

$$V^0 = W^l \quad (12a)$$

$$(1 + \alpha_k \bar{\lambda}) V^k = V^0 + \alpha_k \bar{\lambda} V^{\overline{k-1}} + \alpha_k \Delta \tau L_{irs}^{-1} \bullet \left[ R^{\overline{k-1}}(V) + f \right] \quad k = 1, 2, 3, \dots, K \quad (12b)$$

$$W^{l+1} = V^K \quad (12c)$$

where  $L_{irs}$  denotes the implicit-residual smoothing operator, and  $f$  denotes the multigrid forcing function (which is zero on the finest grid).

The usual coarse-grid equation that would result from applying multigrid to equation (8) is

$$W_\tau^{(2h)} = R^{(2h)}(W) + \left[ I_{(h)}^{(2h)} R^{(h)}(W^{(h)}) - R^{(2h)}(I_{(h)}^{(2h)} W^{(h)}) \right] \quad (13)$$

where the superscript in parentheses denotes the multigrid grid level at which the operator or variable is defined ( $h$  on the fine grid,  $2h$  on the next coarse grid, etc.), and the operator  $I_{(h)}^{(2h)}$  denotes the restriction

process from the fine grid ( $h$ ) to the coarse grid ( $2h$ ). Because  $E$  is invariant during the multigrid cycle, its influence in the operator  $R^{(2h)}$  cancels on the coarse grid. Consequently, the coarse-grid residual can be rewritten to exclude  $E$ , which eliminates the need to restrict and store this term. The actual coarse-grid equation is now

$$W_\tau^{(2h)} = \tilde{R}^{(2h)}(W) + \left[ I_{(h)}^{(2h)} \tilde{R}^{(h)}(W^{(h)}) - \tilde{R}^{(2h)}(I_{(h)}^{(2h)} W^{(h)}) \right] \equiv \tilde{R}^{(2h)}(W) + f^{(2h)} \quad (14)$$

where

$$\tilde{R}^{(m)} = \begin{cases} R(W) & (m) = h \\ S(W) - \bar{\lambda}W & (m) = 2h, 4h, \dots \end{cases} \quad (15)$$

## STABILITY ANALYSIS

Two stability issues are associated with the iterative-implicit method. The first issue is the stability of the implicit equation that contains all the physics (equation (6)). The second issue is the stability and convergence of the iterative algorithm that is used to invert the implicit equation. The second issue can easily be studied independently of the first. The first issue can be studied independently of the second by assuming that the implicit equation can be solved exactly (i.e., the iterative procedure is convergent). The following analysis concentrates on the stability of the Runge-Kutta/multigrid method that is used to solve the implicit equation; however, the section is concluded with a few comments that pertain to the stability of equation (6).

Fourier stability analysis is used to illustrate the effect of treating the physical time derivative implicitly instead of explicitly, as well as to illustrate other algorithmic choices. The analysis is performed for the Runge-Kutta method given by equation (12) with a scalar model equation of the form

$$\frac{\partial W}{\partial \tau} + \bar{\lambda}W = a \left( \frac{\partial W}{\partial x} - \frac{\varepsilon_4}{32} (\Delta x)^3 \frac{\partial^4 W}{\partial x^4} \right) \equiv S_c + S_d \quad (16)$$

The fourth derivative and its scaling closely model the numerical dissipation common to codes such as TLNS3D. Note that because the terms  $E$  and  $f$  are constant during the Runge-Kutta integration, they have no influence on the stability and have been dropped from the analysis. The particular version of Runge-Kutta used by TLNS3D and for this analysis is a five-stage method defined by  $\{\alpha_k\} = \{1/4, 1/6, 3/8, 1/2, 1\}$ . The convection terms are commonly treated differently from the dissipation with regard to the definition of the  $\bar{k}-1$  index. In addition, the  $V^{\bar{k}-1}$  terms, which appear twice on the right-hand side of equation (12b), may be treated differently in each instance. In this stability analysis, the convective and dissipative terms are treated exactly as TLNS3D treats them in a steady-state case. These algorithmic choices are denoted by rewriting (12b) as

$$\begin{aligned} (1 + \gamma\alpha_k\bar{\lambda})V^k &= V^0 + \gamma\alpha_k\bar{\lambda}V^{\bar{k}-1} + \alpha_k\Delta\tau L_{irs}^{-1} \bullet \left( S_c^{k-1} + S_d^{\bar{k}-1} - \bar{\lambda}V^{\bar{k}-1} \right) \\ S_d^{\bar{k}-1} &= \beta_k S_d^{k-1} - (1 - \beta_k) S_d^{\bar{k}-2} \\ S_d^0 &= 0 \end{aligned} \quad (17)$$

where  $\{\beta_k\} = \{1.0, 0.0, 0.56, 0.0, 0.44\}$ . The new parameter  $\gamma$  allows both implicit ( $\gamma = 1$ ) and explicit ( $\gamma = 0$ ) treatments of the physical time derivative to be studied. The choice of  $V^{k-1}$  and  $V^{k-1}$  is the subject of this analysis; however, the analysis will focus on forms that are easily implemented within the current version of TLNS3D. To this end, both  $V^{k-1}$  and  $V^{k-1}$  are defined in a manner similar to the dissipative terms with the use of  $\alpha_k$  and  $\sigma_k$ , respectively.

An evaluation of the spatial derivative with central-difference operators and the transformation to Fourier space gives

$$G^k = \frac{1 + \gamma\alpha_k\bar{\lambda}G_\alpha^{k-1} + \alpha_k \frac{[i\lambda \sin(\theta)G_\beta^{k-1} - \lambda \frac{1}{2}\varepsilon_4 \sin^4(\theta/2)G_\beta^{k-1} - \bar{\lambda}G_\sigma^{k-1}]}{1 + \varepsilon_2 \sin^2(\theta/2)}}{(1 + \gamma\alpha_k\bar{\lambda})} \quad (k = 1, 2, 3, 4, 5) \quad (18)$$

where

$$\begin{aligned} G_\beta^{k-1} &= \beta_k G^{k-1} - (1 - \beta_k)G_\beta^{k-2} \\ G_\alpha^{k-1} &= \alpha_k G^{k-1} - (1 - \alpha_k)G_\alpha^{k-2} \\ G_\sigma^{k-1} &= \sigma_k G^{k-1} - (1 - \sigma_k)G_\sigma^{k-2} \\ G^0 &= G_\beta^0 = G_\alpha^0 = G_\sigma^0 = 1 \\ G_\beta^{-1} &= G_\alpha^{-1} = G_\sigma^{-1} = 0 \end{aligned} \quad (19)$$

$\varepsilon_2$  is the coefficient of implicit-residual smoothing, and  $\lambda \equiv a\Delta\tau/\Delta x$  is the CFL number based on the pseudo-time step.

The influence of implicit versus explicit treatment of the physical time derivative is best illustrated by considering a simplified case of equation (18). Consider the case in which  $\varepsilon_2 = 0$  and  $\alpha_k = \sigma_k = \beta_k$ , for which equation (18) becomes

$$G^k = \frac{1 + \alpha_k \left\{ i\lambda \sin(\theta)G^{k-1} - [\lambda \frac{1}{2}\varepsilon_4 \sin^4(\theta/2) + \bar{\lambda}(1 - \gamma)]G_\beta^{k-1} \right\}}{(1 + \gamma\alpha_k\bar{\lambda})} \quad (k = 1, 2, 3, 4, 5) \quad (20)$$

Equation (20) clearly shows that an explicit treatment of the physical time derivative ( $\gamma = 0$ ) simply translates the stability region to the right as  $\bar{\lambda}$  increases from 0; an implicit treatment reduces the amplification factor, which expands the stability region. This difference is illustrated in figures 1a–c, which show equally spaced contours of the amplification factor  $\|G^5\|$  as a function of the real (dissipative) and imaginary (convective) parts of the spatial operator  $Z(\theta) = \lambda[i \sin(\theta) - \frac{1}{2}\varepsilon_4 \sin^4(\theta/2)]$ . Values of the contour lines are indicated by line types as indicated in the figure legend. Figure 1a shows the steady-state case  $\bar{\lambda} = 0$  as a point of reference. Figures 1b and 1c show the explicit and implicit cases, respectively, for  $\bar{\lambda} = 1$ . Each figure also shows the locus of the spatial operator for  $\lambda = 3$ . Other choices for  $\varepsilon_2$ ,  $\alpha_k$ , and  $\sigma_k$  give qualitatively similar results. An explicit treatment of the physical time derivative will always become unstable for sufficiently large values of  $\bar{\lambda}$ ; the implicit approach is stable for all values of  $\bar{\lambda}$ .

By plotting the amplification along the locus (figure 2), an unusual property of equation (16) is revealed. For  $\bar{\lambda} = 0$ , the amplification goes to 1 as  $\theta$  goes to 0, which is often considered a consistency condition. However, for  $\bar{\lambda} \neq 0$ , the solution is damped across the entire spectrum. This property is a consequence of the source term  $\bar{\lambda}W$ , which appears in both equations (8) and (16) and is not caused by an inconsistency in the derivative operators. The acceleration technique known as enthalpy damping makes use of the same property to improve the convergence of inviscid flows. This analysis suggests

that an implicit treatment of enthalpy damping may lead to further improvements; however, verification of this possibility is beyond the scope of this work.

Alternative choices for  $\alpha_k$  and  $\sigma_k$  are considered with the implicit-residual smoothing reinstated with a coefficient of 0.25. The parameters are tuned to obtain good high-frequency damping, but also to obtain a scheme for which  $\|G^5\|(\theta) > \|G^5\|(2\theta)$  whenever possible. The latter criterion will reduce the influence that aliasing error may have on the coarse-grid equation. Other obvious choices for  $\alpha_k$  and  $\sigma_k$  (in addition to  $\alpha_k = \sigma_k = \beta_k$  used above) are  $\{1.0, 0.0, 0.0, 0.0, 0.0\} \equiv S0$  and  $\{1.0, 1.0, 1.0, 1.0, 1.0\} \equiv S1$ . The first choice corresponds to the case in which  $\widehat{V^{k-1}}$  or  $\widehat{V^{k-1}} = V^0$ ; the second corresponds to the case in which  $\widehat{V^{k-1}}$  or  $\widehat{V^{k-1}} = V^{k-1}$ . Figure 3 shows the amplification for several implicit schemes, where, for example,  $\{S1, \beta_k\}$  denotes that  $\{\alpha_k\} = S1$  and  $\{\sigma_k\} = \{\beta_k\}$ . Although the case with  $\{S1, \beta_k\}$  has the lowest overall amplification, the case with  $\{S1, S1\}$  is more monotone in  $\theta$  and, as such, is the preferred method. Figure 4 shows the amplification of the latter case for a range of  $\bar{\lambda}$ .

So far, the discussion on stability has been limited to the behavior of the Runge-Kutta used to solve equation (6), which does not imply that equation (6) is stable. Equation (6) falls into the class of multistep schemes for which the usual notion of absolute stability is not sufficient to ensure convergence. Instead, the scheme must satisfy the more stringent conditions of *relative stability* [13] to ensure convergence to the proper solution. Equation (6) can be shown to be unconditionally stable when the time operator is approximated to either first or second order. Similarly, time operators of the form given in equation (6) for which  $M$  is also the order of the operator are unconditionally unstable for  $M > 5$ . Although conditionally stable methods would not normally be considered appropriate for large  $\Delta T$  calculations, the nature of the instability is such that even the conditionally stable methods are useful in many situations. A detailed study is beyond the scope of this work; however, the interested reader is referred to the cited reference.

## NUMERICAL RESULTS

To demonstrate the capability of the present method, the results of several numerical experiments are given. The first case that is examined is the solution of an impulsively accelerated flat plate, which is also known as Stokes first problem [14]. An analytic solution for incompressible flow is available for this problem, which allows comparison with solutions obtained numerically with global minimum time stepping (GMTS) and the present method (with variations in time-step size and in the temporal order of the numerical discretization).

The analytic solution of Stokes first problem [14] shows that the time-dependent solution collapses to a single solution of nondimensional velocity versus the similarity parameter  $\eta$  defined as

$$\eta = \frac{y}{2\sqrt{\nu T}} \quad (21)$$

where  $y$  is the direction normal to the flat plate,  $\nu$  is the kinematic viscosity, and  $T$  is the physical time. Figure 5a shows the analytic solution plotted in the similarity parameters. A solution calculated with GMTS after 2000 time steps is also plotted. Calculations were also performed with the present method with 1, 5, and 10 time steps to reach the same physical time as the GMTS solution. Different orders



of accuracy of the discretization of the physical time derivative were used, and the solutions are plotted in figures 5b–f.

Figure 5b shows the comparison of a first-order solution for the various time steps with the GMTS solution. The single time step does not accurately match the GMTS. As more time steps are used (smaller physical time steps), the comparison improves, although all show some error from  $\eta = 1.5$  to  $\eta = 2.0$ .

In figure 5c, comparisons of the present method with second-order physical time discretizations for the three time-step sizes are made with the GMTS. As expected, the smaller the time step, the better the agreement.

Comparisons for third-, fourth-, and fifth-order discretizations are shown in figures 5d–f, respectively. For the third-order solution, good agreement occurs for the two smaller time steps; however, this agreement degrades for the fourth- and fifth-order solutions. This degradation can be attributed to starting errors caused by the unavailability of information at previous time steps for the first few steps of the higher order schemes. This study demonstrates that third-order discretization agrees with the GMTS solution better than second-order discretization.

The work units required to obtain the solution of Stokes first problem at the same physical time as the GMTS solution after 2000 steps are shown in table 1 for a first- through a fifth-order physical time discretization. The single step solutions with the present method were performed in the least number of work units; however, the accuracy of the solution is unacceptably poor. For first- and second-order time discretizations, the ten-step solutions were obtained in fewer work units than the five-step solutions because of better convergence. (All work units in the table are based on converging the viscous drag at each time step to six significant digits.) Third- through fifth-order solutions required fewer work units for the five-step calculations than for the ten-step calculations. These results suggest that a balance exists between convergence speed at a given time step and the number of time steps used to obtain a solution with the present method. If the time step is too large, then the accuracy is poor. If the time step is too small, then unnecessary work is expended to converge at unneeded time steps.

Table 1. – Work units required to calculate solution for Stokes first problem.

Order of Physical Time Discretization	Number of Physical Time Steps		
	1	5	10
1st	54.6	210.9	193.8
2nd	51.2	182.4	171.0
3rd	42.1	159.5	171.0
4th	15.8	165.3	193.8
5th	15.8	153.8	171.0
GMTS	2000.0		

The second test case used to demonstrate the present method is the unsteady flow over an impulsively started two-dimensional circular cylinder (with a Reynolds number of 1200 and a Mach number of 0.3). Detailed experimental and numerical investigations of the flow behind a cylinder have been performed previously by other authors. (See, for example, reference [15].) The initial flow is symmetric with zero

lift as the wake behind the cylinder begins to grow. As the wake continues to grow, it becomes unstable and begins to shed from alternate sides of the cylinder.

An examination is made of the first part of the solution where the symmetric wake begins to grow. In these calculations, 4600 GMTS steps were used to reach  $t^* = 2.4$ . Detailed comparisons of the first-through fourth-order solutions with both the present method and GMTS for  $t^* = 0$  to  $t^* = 2.4$  are shown in figure 6. The calculations were performed with the present scheme with 10, 20, and 40 steps to reach  $t^* = 2.4$ . As with Stokes first problem, the first-order discretization does not give satisfactory results (figure 6a). Second-order differencing (figure 6b) shows much better agreement with the GMTS solution. Third-order differencing shows good agreement for the two smaller time-step sizes (figure 6c). Fourth-order differencing gives bad overshoots for the 10-step calculation and instability for the 20- and 40-step calculations (figure 6d).

The present scheme was then used to calculate the flow around the cylinder out to times where the vortex shedding occurred. Time histories of the lift coefficient  $C_l$  and the drag coefficient based on integrated pressures  $C_{d_p}$  are shown in figure 7. From experimental data and the results of the GMTS calculations shown in reference [15], the period of the oscillation of  $C_{d_p}$  is known to be approximately 4 in terms of the nondimensional time  $t^*$ . To give 40 time steps per period, a time step of  $\Delta t^* = 0.1$  was used. This time-step size is roughly equal to the time step used in the 20-step calculations shown in figure 6. The first-order discretization predicted a Strouhal number of 0.21. The second- and third-order discretizations predicted a Strouhal number of 0.24 compared with the experimentally obtained value of 0.21. The fourth-order physical time discretization calculation diverged.

## CONCLUSIONS

A method to accurately calculate time-accurate solutions to the unsteady Navier-Stokes equations has been presented. Multigrid acceleration has been successfully employed to accelerate the calculations of the *iterative-implicit* method. Run times that are one order of magnitude smaller than the run times required for global minimum time stepping have been demonstrated.

# FIGURES

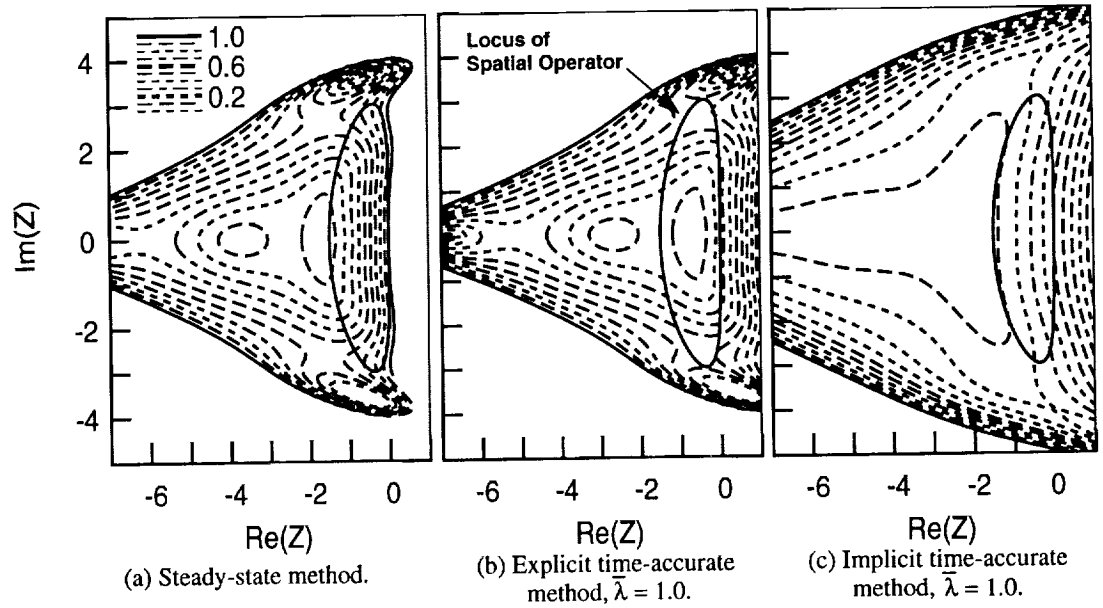


Figure 1. Contours of stability region.

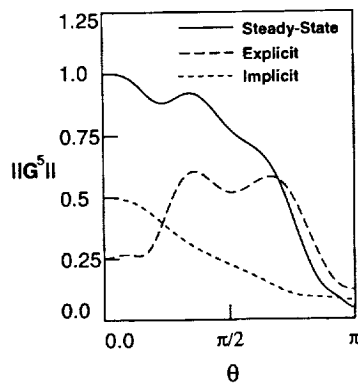


Figure 2. Amplification along the locus of the spatial operator.

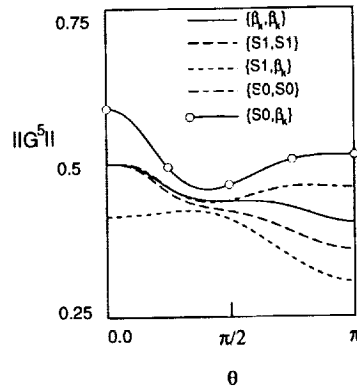


Figure 3. Amplification of implicit schemes with  $\bar{\lambda} = 1.0$ .

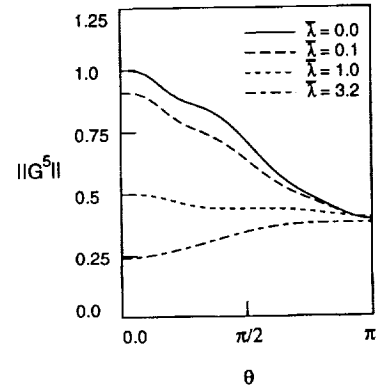


Figure 4. Amplification of  $\{S1, S1\}$  case for a range of  $\bar{\lambda}$ .

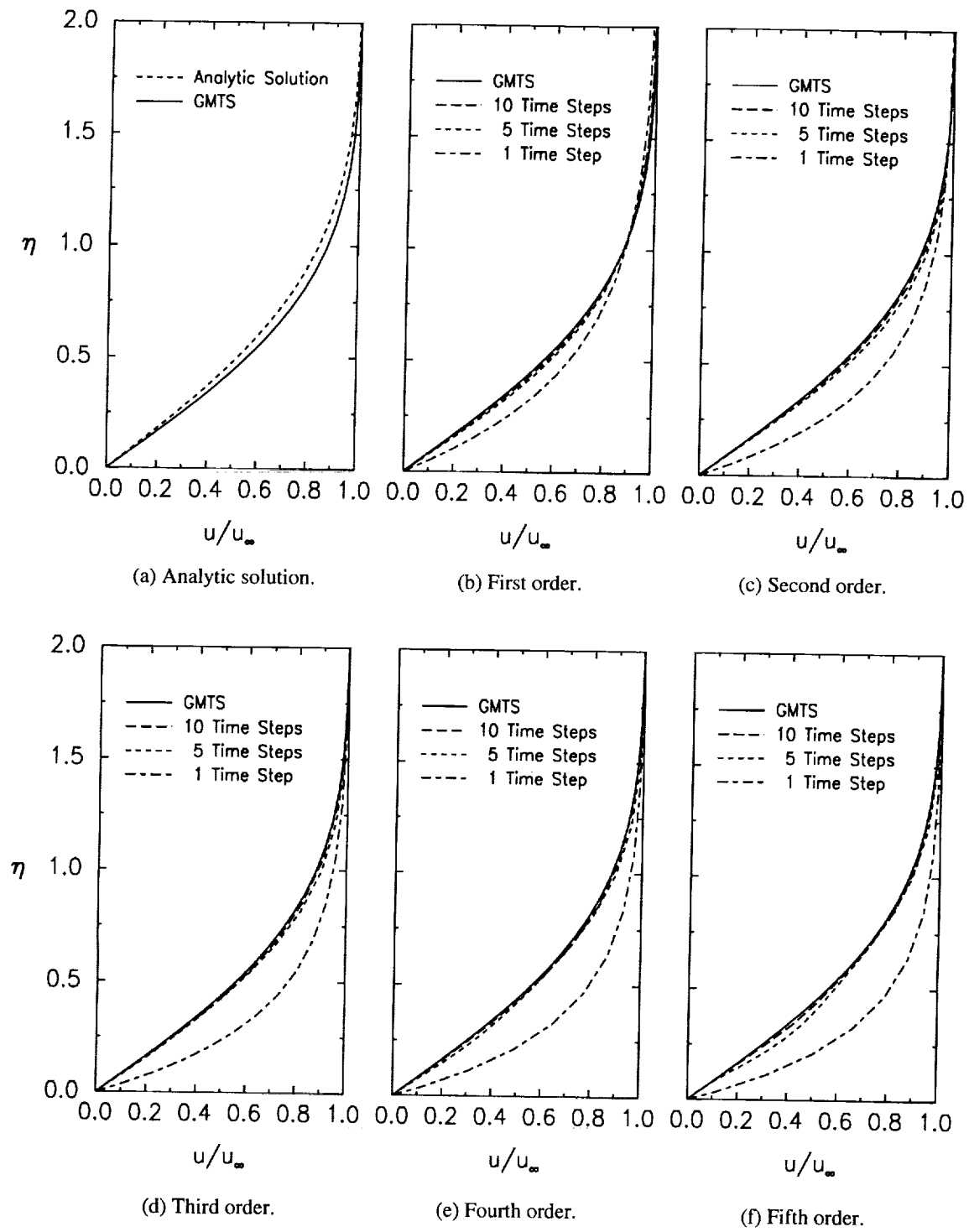


Figure 5. Velocity profiles for impulsively started flat plate.

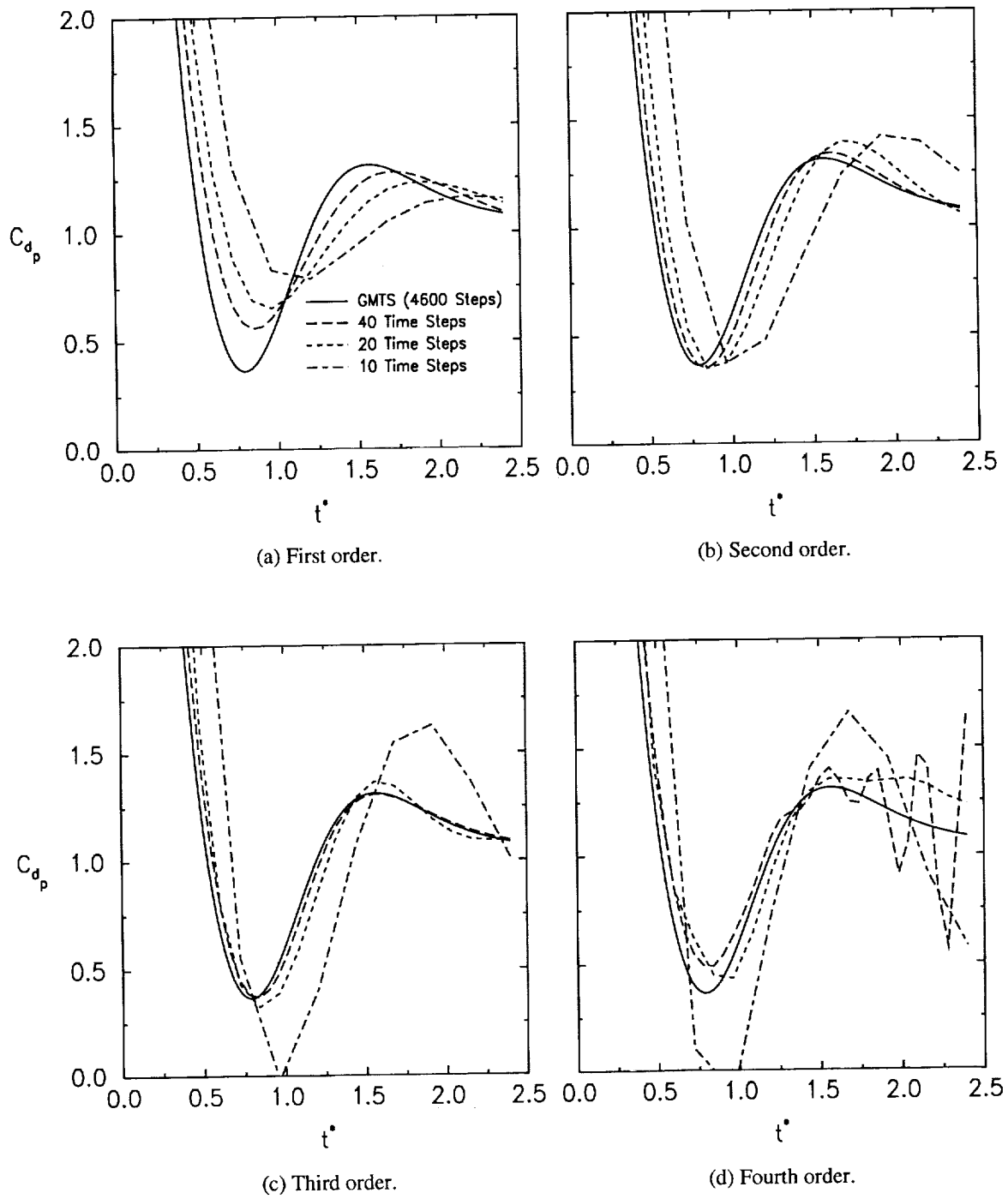
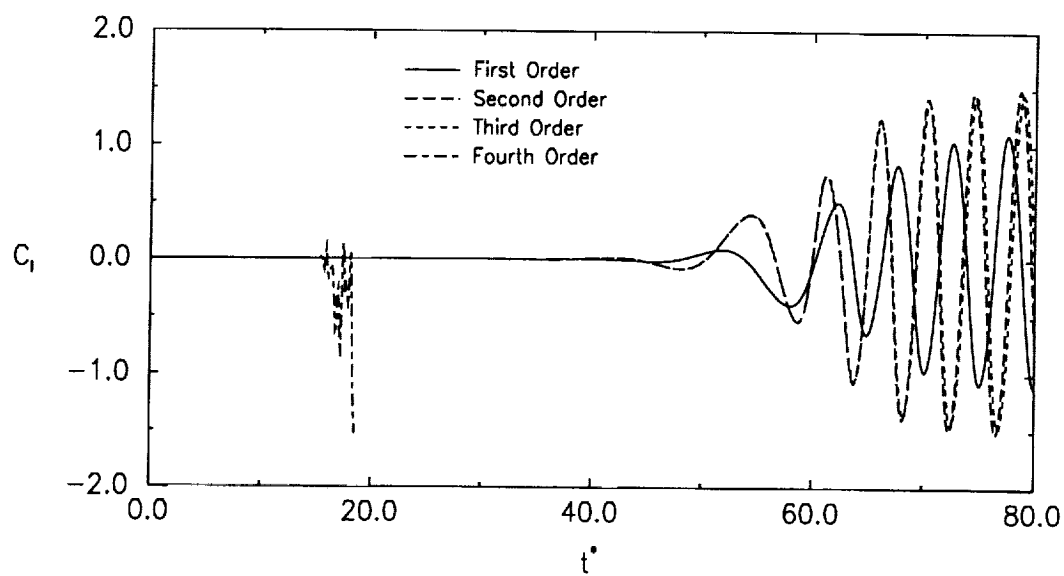
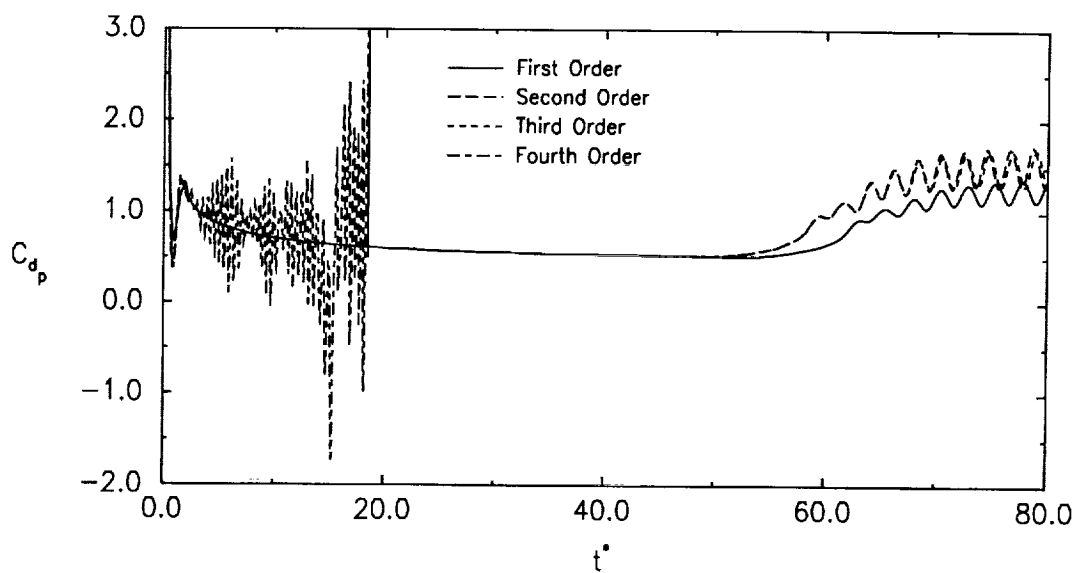


Figure 6. Initial pressure drag-coefficient history for impulsively started cylinder.



(a) Lift history.



(b) Pressure drag history.

Figure 7. Lift- and drag-coefficient histories for impulsively started cylinder.

## REFERENCES

- [1] V. N. Vatsa and B. W. Wedan. Development of an Efficient Multigrid Code for 3-D Navier-Stokes Equations and its Application to a Grid-Refinement Study. *Computers and Fluids*, 18(4):391-403, 1990.
- [2] F. Ghaffari, J. M. Luckring, J. L. Thomas, and B. L. Bates. Navier-Stokes Solutions About the F/A-18 Forebody-Leading-Edge Extension Configuration. *Journal of Aircraft*, 27:737-748, 1990.
- [3] H. L. Atkins. A Multi-Block Multigrid Method for the Solution of the Euler and Navier-Stokes Equations for the Three-Dimensional Flows. AIAA Paper 91-0101, 1991.
- [4] A. Jameson. Time Dependent Calculations Using Multigrid, with Applications to Unsteady Flows Past Airfoils and Wings. AIAA Paper 91-1596, 1991.
- [5] III Arthur C. Taylor, Wing-Fai Ng, and Robert W. Walters. Upwind Relaxation Methods for the Navier-Stokes Equations Using Inner Iterations. *Journal of Computational Physics*, 99:68-72, 1992.
- [6] B. S. Baldwin and H. Lomax. Thin Layer Approximation and Algebraic Model for Separated Turbulent Flows. AIAA Paper 78-257, 1978.
- [7] A. Jameson, W. Schmidt, and E. Turkel. Numerical Solutions of the Euler Equations by Finite Volume Methods Using Runge-Kutta Time-Stepping Schemes. AIAA Paper 81-1259, 1981.
- [8] A. Jameson and T. J. Baker. Solutions of the Euler Equations for Complex Configurations. AIAA Paper 83-1929, 1983.
- [9] E. Turkel and V. N. Vatsa. Effect of Artificial Viscosity of Three Dimensional Flow Solutions. AIAA Paper 90-1444, 1990.
- [10] L. Martinelli. *Calculation of Viscous Flows with Multigrid Methods*. PhD thesis, MAE Dept., Princeton University, 1987.
- [11] R. C. Swanson and E. Turkel. Artificial Dissipation and Central Difference Schemes for the Euler and Navier-Stokes Equations. AIAA Paper 87-1107, 1987.
- [12] V. N. Vatsa, M. D. Sanetrik, and E. B. Parlette. Development of a Flexible and Efficient Multigrid-Based Multiblock Flow Solver. AIAA Paper 93-0677, 1993.
- [13] C. William Gear. *Numerical Initial Value Problems in Ordinary Differential Equations*. Prentice-Hall, Inc., 1971.
- [14] Hermann Schlichting. *Boundary Layer Theory, Seventh Edition*. McGraw-Hill Book Company, 1979.
- [15] Christopher L. Rumsey. Details of the Computed Flowfield Over A Circular Cylinder At Reynolds Number 1200. Presented at the Forum on Unsteady Flow Separation, Cincinnati, Ohio, June 15-17, 1987.

